

PATENT

Please replace the paragraph beginning at page 13, line 6, with the following rewritten paragraph:

-- A genealogy is a tree structure of versions where leaves represent the latest-and-greatest versions, and all other versions are kept for historical, checkpoint, release correlation, and component sharing purposes. There is more than one leaf version only when there is more than one lineage. In the example, there are two lineages: A and B. Each leaf, and its lineage, represents the evolution of a separate and distinct configuration item (CI). In prior art ODBMS, such as Objectivity, a well-known commercial of the shelf ODBMS, with supported versioning, a lineage is created using linear versioning and a new lineage is introduced using branch versioning. --

Please replace the last paragraph on page 14 with the following rewritten paragraph:

-- The genealogy starts when CP-1 CP1 creates A1.CP1.1, sets all values as desired, and publishes it as A1, an initial version. --

Please replace the three paragraphs beginning at the top of page 15, and also the first paragraph on page 16, with the following rewritten paragraphs:

-- The genealogy is extended when CP-2 CP2 creates a version of A1 (A2.CP2.1), makes changes to A2.CP2.1, and then publishes it as A2. Publication in this case entails making a version that is a copy of A1 – the latest in the genealogy at this point, and then applying changes made within the CP. Changes are determined by calculating the difference between A1 and A2.CP2.1.

PATENT

– Both ~~CP-3 CP3~~ and ~~CP-4 CP4~~ use A2 as their change start source. (This is now true concurrency; both are effectively changing the same version (A2) at the same time without affecting one another – at least until publication time.) Since ~~CP-3 CP3~~ is the first to publish, its process is the same as was for ~~CP-2 CP2~~. But ~~CP-4 CP4~~ has a problem; ~~CP-3 CP3~~ has created A3, which is different from ~~CP-4's CP4's~~ change start source. If ~~CP-4 CP4~~ were to publish in the same manner as ~~CP-3 CP3~~ had, then ~~CP-3's CP3's~~ changes would be lost; A4 would replace A3 as the latest-and-greatest, but not contain any of the changes contributed by A3.

– The solution is to first merge A3 (containing ~~CP-3's CP3's~~ changes) to ~~CP-4 CP4~~. That way, ~~CP-3's CP3's~~ changes will be preserved when ~~CP-4 CP4~~ publishes. A3.CP4.1 is made into a checkpoint version, and merge changes are applied to a new version (A4.CP4.2). A checkpoint version is necessary here because the purpose for changes has changed; originally, changes were made according to the CP's rationale, but now they are being made to perform a merge.

– Merging is accomplished by comparing the changes made from the change start source version to the latest-and-greatest (A2 to A3) with the changes made in the CP (A2 to A3.CP4.1). Identical changes or those affecting different areas are considered trivial and are applied to A4.CP4.2 automatically. Changes not in agreement are considered non-trivial and require user-directed resolution. For each non-trivial change, the user must choose one of: the change start source's (A2) value, the latest-and-greatest version's (A3) value, the CP's (A3.CP4.1) value, or provide a new value. For historical purposes, the user's decision is recorded as a resolution object attached to the merge target (A4.CP4.2). --

PATENT

Please replace the second paragraph on page 17, beginning on line 8, with the following rewritten paragraph:

– The situation expressed by ~~CP-5~~ CP5 and ~~CP-6~~ CP6 looks similar to the merge situation of ~~CP-3~~ CP3 and ~~CP-4~~ CP4 in that both ~~CP-5~~ CP5 and ~~CP-6~~ CP6 have the same change start source, but is very different because ~~CP-6~~ CP6 has declared its version (B1.CP6.1) to be an initial version. As a result, publication of ~~CP-6~~ CP6 creates B1 in the genealogy as a derivative of A4, rather than the next version of A4. Like A1 originally created in CP1, B1 begins a lineage that is separate and distinct from all others. Contributions to A's lineage, like that of ~~CP-5's~~ CP5's A5, are of no concern to the new B lineage and do not need to be merged. –

Please replace the paragraph beginning at page 18, line 8, with the following rewritten paragraph:

– Now referring to Figure 2, there is shown a ~~CP-4~~ CP4 used in the genealogy evolution example, of the system and method of the present invention, but expanded to show that a CP can affect multiple CI lineages; in the example: C, H, P, and Z, in addition to A. –

Please replace the paragraph beginning at page 19, line 20, with the following rewritten paragraph:

– At initial checkout time, the user chooses whether or not the new version begins a new lineage. If so, it is derived from the source version instead of becoming a new latest of it. If not, the decision to have a version start a new lineage can still be made up to and including the point of merging, where the decision might be used to

PATENT

resolve merge conflicts. The one caveat, though, is that once a version is made the start of a new lineage, there is no easy way back. The only recourse is for the user to move all current uses of the new lineage to the old, delete the new lineage, and re-checkout as a ~~latest-and-greatest~~. —

Please replace the paragraph beginning at page 20, line 12, with the following rewritten paragraph:

— With respect to Figures 4-9 discussed in detail below:

Configuration Items

Configuration items (CIs) are objects under configuration control and come in two ~~flavors~~ varieties: model and model folder. A model correlates to a typical CM system's file, and a model folder correlates to a typical CM system's directory (or folder). —

Please replace the first and third paragraphs beginning at the top of page 24 under "Views" with the following rewritten paragraphs:

— Views

— The job of a view is to select one and only one version from each genealogy in the database. For the public view, all the ~~latest-and-greatest~~ in the genealogy are chosen. For a CP view, the latest checkpoint version is chosen for checked-out CIs and the ~~latest-and-greatest~~ in the genealogy for those not checked out. —

— There is but a single lineage of Top folders, the ~~latest-and-greatest~~ providing the public view. Each CP gets its own checked-out version of the ~~latest-and-greatest~~ Top folder when it is started, providing each CP a private view. —